REMARKS

Reconsideration and allowance of this application are respectfully requested.

Claims 1-65 are pending in this application.

As described in the present application, a "typical content delivery network (CDN) operator deploys one or more parent servers, hosting a plurality of objects, in a network and one or more edge servers at the edge of the network to facilitate more cost-effective and efficient delivery of such objects to an end-user (client)." ¶0010. A block diagram of an exemplary topology of the managed object replication and delivery method and system according to embodiments of the invention is shown in Figure 1 of the application (reproduced below).

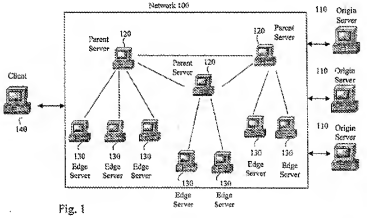

Fig. 1

In one aspect, this invention is a method for managed object replication and delivery. This method is described in the application as filed, e.g., with reference to Fig. 2 (replicated below) which "depicts embodiments of the method in relation to a portion of the network 100, an origin server 110 and a client 140 . . ." ¶0023.

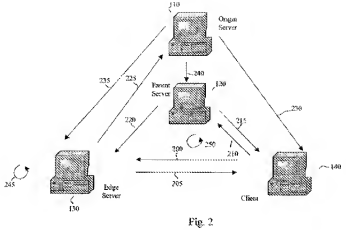

Fig. 2

"Initially, the method . . . directs . . . a client, requesting one or more objects, to an edge server in the network, whether or not the edge server has the requested object(s)." ¶0024. "The selected . . . edge server 130 determines . . . whether the edge server already has the requested object and, if so, serves . . . the object to the requesting client 140." ¶0025. "If the selected edge server does not have the requested object, a check is initiated . . . for the edge server to determine whether the requested object is popular and if so, to replicate the popular requested object to the edge server." ¶0026. "Further, if the selected edge server does not have the requested object, the selected edge server directs . . . the requesting client 140 to a parent server 120. Preferably the client 140 is redirected to a parent server that has the requested object and is able to serve . . . the requested object to the client." ¶0028.

So, e.g., as recited in claim 1 and its dependents (claims 2-14), the method includes: Directing a request by a client for an object to an edge server in a network. If the edge server has the requested object, the requested object is served to the client. Otherwise (i.e., if the edge server does *not* have the requested object), the client request is redirected to a server that has the requested object, and the requested object is served to the client. If the requested object is popular, the requested object is replicated to the edge server.

Similarly, claims 23-37 recite a computer program product including computer program code to cause a processor to perform the methods for managed object replication and delivery of claims 1-15. Independent claim 16 recites a method similar to independent claim 1, for managed object replication and delivery. Independent claim 16 (and its dependents) recites "directing a request by a client for an object to *an optimal* edge server in a network." Claim 38 recites a computer program product including computer program code to cause a processor to perform the method of claim 16. Independent claim 45 recites a system for managed object replication and delivery including a plurality of edge servers in a network; and a plurality of parent servers in the network. As recited in the claim, at least one of the plurality of edge servers and the plurality of parent servers: direct a request by a client for an object to an edge server in the network, if the edge server has the requested object, serve the requested object to the client, otherwise, redirect the client request to a server that has the requested object and serve the requested object to the client, if the requested object is popular, replicate the requested object to the edge server. Independent claim 59 is similar to claim 45, and recites "at least one of the plurality of edge servers and the plurality of parent servers: direct a request by a client for an object to an optimal edge server in the network."

THE PRIOR ART REJECTIONS

The Examiner has rejected claims 1-65 under 35 U.S.C. § 103 as being unpatentable over Jungck and Sim. The grounds for this rejection are respectfully traversed.

The Examiner states that "[a]s per claims 1, 16, 23, 38, 45 and 59, Jungck disclosed a system . . . comprising: a plurality of edge servers in a network; and a plurality of parent servers in the network (paragraph 19), wherein at least one of the plurality of edge servers and the plurality of parent servers (paragraph 25): direct a request by a client for an object to an edge server in the network

(paragraphs. 27 & 35), if the edge server has the requested object, serve the requested object to the client (paragraph 56), otherwise redirect the client request to a server that has the requested object and serve the requested object to the client (paragraph 57)." Applicant respectfully disagrees.

In Jungck, if a client request is sent to a server which does not have the requested object, that server tries to obtain the object. The server does not, as claimed, redirect the client request to a server that has the requested object.

The Examiner argues (citing Jungck paragraph 57) that in Jungck, if the edge server does not have a requested object, then it redirects the client request." However, Jungck neither teaches nor in any way suggests redirecting a request for an object from an edge server when that edge server does not have the requested object.

Jungck (in paragraph 0057) describes the operation of so-called cache servers. "Cache servers 208 invisibly intercept requests for content and attempt to provide the requested content from the cache (also known as a "hit"). . . . Where the requested content is not in the cache . . ., *the cache* ***forwards*** *the request onto the content source*. When the source responds to the request by sending the content to the client 102, 104, 106, the cache server 208 saves a copy of the content in its cache for later requests. In the case where a cache server is part of a proxy server, *the cache/proxy server makes the request to the source on behalf of the client* 102, 104, 106. *The source then provides the content to the cache/proxy server which caches the content and also forwards the requested content to the client 102, 104, 106*.", *Jungck* ¶0057, emphasis added.

So, what one of Jungck's cache servers does is this: if it has the requested content then it tries to serve that content. If it does not have the requested content then it gets the content from the source and then serves it to the requesting client.

Thus Jungck lacks at least one element of the claims.

The Examiner acknowledged that Jungck does not teach anything about testing for popularity of requested objects.

In order to try to overcome this deficiency in Jungck, the Examiner applies Sim. Sim describes so-called "distribution stations" at which he keeps all or part of large objects. In Sim, "[e]ach distribution station is configured to determine how much of the content to save locally, based on information such as usage, popularity, etc." ¶0047 and "the portions and amounts of a large payload file maintained at each node depends on the available storage, popularity of the content, distribution criteria by the content provider, etc." ¶0052.

However, Sim does not, as the Examiner would have it, teach "that if the requested object is popular, replicate the requested object to the edge server."

Accordingly, applicant respectfully submits that no proposed combination of Jungck and Sim would produce the presently claimed invention.

Further as to claims **2, 24 and 46**, applicant respectfully submits that no proposed combination of Jungck with Sim would teach or in any way suggest the claimed: "redirecting the client request to a parent server in the network that has the requested object and serving the requested object to the client from the parent server." First, neither Jungck nor Sim has any notion of parent servers. Second, in Jungck, failed requests are handled by the cache servers, not by redirecting requests anywhere, let alone to parent servers. For at least this reason, these claims are further patentable over any proposed combination of Jungck with Sim.

Further as to claims **3, 25 and 27**, applicant respectfully submits that no proposed combination of Jungck with Sim would teach or in any way suggest the claimed: "redirecting the client request to a parent server in the network that does not have the requested object, *recursively redirecting* the request until a parent server in the network having the requested object is reached and serving the requested object to the client from the parent server." Neither Jungck nor Sim has any teaching or suggestion of any kind of recursive redirecting of requests. For at

least this reason, these claims are further patentable over any proposed combination of Jungck with Sim.

Further as to claims **13, 19, 35, 41, 56 and 61**, applicant respectfully submits that no proposed combination of Jungck with Sim would teach or in any way suggest the claimed: "replicating the requested object in accordance with a dynamic replication threshold." Sim (in ¶0230, relied upon by the Examiner) teaches replacing content based on popularity (less popular content is replaced by more popular content). But Sim is completely silent and lacks any teaching or suggestion of anything like the claimed ***dynamic*** replication threshold. For at least this reason, these claims are further patentable over any proposed combination of Jungck with Sim.

The Examiner has also failed to provide any motivation to combine the teachings of Jungck and Sim. Applicant respectfully submits that one of skill in the art would not have combined Jungck with Sim because, *inter alia*, *Jungck teaches away* from caching based on popularity. The only caching "policies" described in Jungck are based on temporal or spatial locality. As described in Jungck, "[c]aches operate on two principles, temporal locality and spatial locality. Temporal locality is a theory of cache operation which holds that data recently requested will most likely be requested again. This [temporal locality] theory dictates that a cache should store only the most recent data that has been requested and older data can be eliminated from the cache. Spatial Locality is a theory of cache operation which holds that data located near requested data (e.g. logically or sequentially) will be likely to be requested next. This theory dictates that a cache should fetch and store data in and around the requested data in addition to the requested data." *Jungck ¶0058.*

In view of the above, withdrawal of this rejection under § 103 is respectfully requested.

CONCLUSION

Applicant respectfully submits that all claims are in condition for allowance and an early action to that effect is earnestly solicited. The Examiner is invited to contact the undersigned at the number provided to resolve any outstanding issues.

CHARGE STATEMENT: Deposit Account No. 501860, order no. **2615-0040.**
The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/ Order Nos. shown above, for which purpose a duplicate copy of this sheet is attached.
This CHARGE STATEMENT does not authorize charge of the issue fee until/unless an issue fee transmittal sheet is filed.

CUSTOMER NUMBER

42624

Respectfully submitted,

By: /Brian Siritzky/ Reg. #37,497
Brian Siritzky, Ph.D.

Davidson Berquist Jackson & Gowdey LLP, 4300 Wilson Blvd., 7th Floor, Arlington, VA 22203
Main: (703) 894-6400 • FAX: (703) 894-6430